

# Approximation Algorithms for the Joint Replenishment Problem with Deadlines<sup>\*</sup>

Marcin Bienkowski<sup>1</sup>, Jaroslaw Byrka<sup>1</sup>, and Marek Chrobak<sup>2</sup>

<sup>1</sup> Institute of Computer Science, University of Wrocław, Poland.

<sup>2</sup> Department of Computer Science, University of California at Riverside, USA.

**Abstract.** The Joint Replenishment Problem (JRP) is a fundamental optimization problem in the supply chain management concerned with optimizing the flow of goods between a supplier and retailers. In JRP, retailers receive demands for a commodity. To satisfy these demands they place orders at the warehouse which, in turn, issues aggregate orders at the supplier. The objective is to compute a schedule of orders that minimizes the sum of the ordering costs and delay costs.

In this paper, we study the approximability of JRP-D, the variant of JRP where there is no delay cost, but the demands needs to be satisfied before specified deadlines. The main result of the paper is an  $e/(e - 1)$ -approximation algorithm based on iterative rounding, improving the previous bound of  $5/3$ . For the case of equal-length demand periods, we improve the ratio to 1.5 and we show that this case remains  $\mathsf{APX}$ -hard even if each retailer has at most four demands.

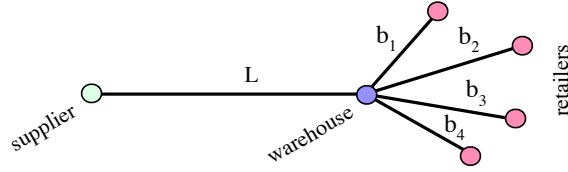
## 1 Introduction

The Joint Replenishment Problem with Deadlines (JRP-D) is an optimization problem in the inventory theory concerned with optimizing the flow of goods between suppliers and retailers. JRP-D arises in a distribution setting with one supplier, one warehouse and  $m$  retailers, cf. Figure 1. The retailers receive demands for a certain commodity that they need to provide within a specified time period. To serve these demands, the retailers place orders at the warehouse which, in turn, places orders at the supplier. As any single order comes with a cost, it is desired to aggregate the orders at each retailer and at the warehouse, while still assuring that the goods reach retailers in time.

More concretely, the problem specification involves a fixed *warehouse ordering cost*  $L$  and, for each retailer  $\rho$ , his *retailer ordering cost*  $b_\rho$ . A *demand* is represented by a triple  $(\rho, r, d)$ , where  $\rho \in \{1, \dots, m\}$  is the identifier of the retailer,  $r \in \mathbb{Q}$  is the demand's release time and  $d \in \mathbb{Q}$  is its deadline. The interval  $[r, d]$  will be referred to as the *demand period*. At any chosen time  $t$ , a subset  $R$  of retailers may place an order, through the warehouse, at the supplier, which is then served immediately and *satisfies* all the demands  $(\rho, r, d)$ , such that  $\rho \in R$ ,

---

<sup>\*</sup> Research supported by MNiSW grant number N N206 368839, 2010-2013 and by NSF grant CCF-1217314.



**Fig. 1.** An illustration of an instance with four retailers, where the ordering costs are represented graphically as distances. The cost of an order is the total weight of the subtree connecting the supplier and the involved retailers.

and  $t \in [r, d]$ . The cost of this order is then equal to the ordering cost of the warehouse plus the ordering costs of respective retailers, i.e.,  $L + \sum_{\rho \in R} b_{\rho}$ . It is convenient to think of this order as consisting of a warehouse order of cost  $L$ , which is then joined by each retailer  $\rho \in R$  at cost  $b_{\rho}$ . A *schedule* is a collection of orders. A schedule is *feasible* if it satisfies all the demands. The objective of the JRP-D problem is to find — for any given set of demands — a feasible schedule that minimizes the total cost.

**Previous Results.** The decision variant of JRP-D problem was shown to be strongly  $\text{NP}$ -complete by Becchetti et al. [3]. (They used a different terminology, considering an equivalent problem of packet aggregation with deadlines on two-level trees.) Later, Nonner and Souza [10] showed that JRP-D is  $\text{APX}$ -hard, even in the special case when each retailer issues only three demands. A 2-approximation LP-based algorithm was given by Levi, Roundy and Shmoys [7]. The approximation ratio was subsequently improved to 1.8 by Levi et al. [6,8,9], using randomized rounding, and then to  $5/3$  by Nonner and Souza [10].

**Our Contribution.** The main contribution of our paper is to further improve the approximation ratio to  $e/(e-1) \approx 1.58$ . Our algorithm is based on a recently developed technique of iterative randomized rounding of an LP-relaxation, thus showing a more general applicability of this approach. We also consider a specific case in which all demand periods are of equal length. From the practical standpoint, this would capture the scenario where the demand period lengths are set globally, for example via industry standards. We show that this version is as hard as the general case, by proving that JRP-D remains  $\text{APX}$ -hard with this restriction, in fact even if each retailer has only four demands. On the other hand, for equal-length demand periods we show a simple 1.5-approximation.

**Related Work.** JRP-D is a sub-case of a more general Joint Replenishment Problem (JRP). In JRP, instead of having a deadline, each demand is associated with a delay cost function that specifies the cost for the delay between the times the demand was released and satisfied by an order. JRP is  $\text{NP}$ -complete, even if the delay function is linear [2,10]. JRP is in turn a sub-case of One-Warehouse Multi-Retailer (OWMR) problem, where the commodities may be stored at the warehouse for a certain cost per time unit. For instance, the 1.8-approximation result by Levi et al. [9] holds also for the general setting of OWMR. JRP was

also studied in the online scenario, where a 3-competitive algorithm was given by Buchbinder et al. [4].

Another straightforward generalization of JRP involves a tree-like structure with the supplier in the root and retailers at the leaves. This setting captures the problem of packet aggregation in convergecasting trees. For this model, a 2-approximation is known for the variant with deadlines [3], and an  $O(\log \theta)$ -competitive online algorithm is known for linear delay costs [5], where  $\theta$  is the sum of all nodes' ordering costs.

## 2 Iterative-Rounding Approximation Algorithm

In this section we present our Algorithm JMM for JRP-D. The algorithm solves the linear program for the relaxation of JRP-D and then rounds the obtained fractional solution  $\mathbf{x}$  to an integral solution. The rounding process is based on the method of randomized iterative rounding. We show that Algorithm JMM computes an order schedule  $\hat{\mathbf{x}}$  whose expected total cost is bounded by  $e/(e-1)$  times the optimal cost.

**LP relaxation.** Without loss of generality, the orders are placed only at starting points of some demand periods. Let  $D$  be the given set of demands, and let  $T = \{r : (\rho, r, d) \in D\}$  be the set of the starting points for all the demands in the instance. The following linear program is the fractional relaxation of the natural integer program for JRP-D.

$$\begin{aligned} \text{minimize} \quad & \text{cost}(\mathbf{x}) = \sum_{t \in T} (Lx_t + \sum_{\rho \in \{1, \dots, m\}} b_\rho x_t^\rho) \\ \text{subject to} \quad & x_t \geq x_t^\rho \quad \text{for all } t \in T, \rho \in \{1, \dots, m\} \quad (1) \\ & \sum_{t \in T, r \leq t \leq d} x_t^\rho \geq 1 \quad \text{for all } (\rho, r, d) \in D \quad (2) \\ & x_t, x_t^\rho \geq 0 \quad \text{for all } t \in T, \rho \in \{1, \dots, m\} \end{aligned}$$

Let vector  $\mathbf{x}$  be an optimal fractional solution to this LP relaxation. We show how to construct a feasible integral vector  $\hat{\mathbf{x}}$  such that  $\text{cost}(\hat{\mathbf{x}}) \leq \frac{e}{e-1} \cdot \text{cost}(\mathbf{x})$ .

**Real time and virtual time.** Let us start with some intuition. We will use fractional orders as an alternative notion of time. Any *real* time  $\tau \in T$  will be associated with the *virtual* warehouse time interval between  $\sum_{t \in T, t < \tau} x_t$  and  $\sum_{t \in T, t \leq \tau} x_t$ . The concept of virtual time of a retailer  $\rho$  can be defined analogously. Since warehouse orders are placed more often than retailers' orders (as encoded in constraint (1)), this virtual time will flow at least as fast at the warehouse as it flows at any retailer. To guarantee feasibility of the obtained solution, it suffices that orders of each retailer happen sufficiently often in his virtual time measure (as encoded in constraint (2)).

Let  $v = \sum_{t \in T} x_t$  denote the (possibly fractional) number of warehouse orders in the solution  $\mathbf{x}$ . Without loss of generality we can assume that the instance contains at least one demand, so  $v \geq 1$ . For  $\alpha \in [0, v]$  define

$$rt(\alpha) = \min \left\{ \tau \in T : \sum_{t \in T, t \leq \tau} x_t \geq \alpha \right\},$$

which is a function that translates the warehouse virtual time into real time. For  $\alpha \in [0, v]$  and  $\rho \in \{1, \dots, m\}$  define the quantity

$$Density_\rho(\alpha) = \frac{x_{rt(\alpha)}^\rho}{x_{rt(\alpha)}},$$

which relates different virtual time notions. If  $x_{rt(\alpha)} = 0$ , then  $x_{rt(\alpha)}^\rho = 0$  as well, and we set  $Density_\rho(\alpha) = 0$  in such case. By constraint (1),  $Density_\rho(\alpha) \leq 1$  for all  $\rho$  and  $\alpha$ . Note that in these terms, the (possibly fractional) number of orders of retailer  $\rho$  in the solution  $\mathbf{x}$  is equal to  $\sum_{t \in T} x_t^\rho = \int_0^v Density_\rho(y) dy$ . We denote this amount by  $v^\rho$ .

**Algorithm JMM.** In our construction, we will use a random number generator  $Rand()$ , which generates random numbers from  $(1/e, 1)$  with a distribution defined by the density function  $1/y$  for  $y \in (1/e, 1)$ . Thus the probability that  $Rand() \in (z, z')$ , for  $1/e < z < z' < 1$ , is equal  $\ln(z'/z)$ . The expected value of this distribution is  $\mathbf{E}[Rand()] = (e - 1)/e$ .

To compute  $\hat{\mathbf{x}}$ , Algorithm JMM first computes the warehouse ordering times, and then, for each retailer  $\rho$ , it determines which orders should  $\rho$  join. The details of the algorithm are given in Pseudocode 1. In the warehouse virtual time, its ordering times are computed (Lines 2–5) as a sequence  $(\alpha_j)_j$ , with the length of each interval  $[\alpha_j, \alpha_{j+1}]$  determined by  $Rand()$ . Then we consider each retailer  $\rho$  separately. For each  $\rho$ , we compute its order times (in the warehouse virtual time) as a sequence  $(\beta_i^\rho)_i$ , which is a sub-sequence of  $(\alpha_j)_j$ . The idea is to spread these times as sparsely as possible (to minimize cost), while ensuring feasibility. After computing each  $\beta_i^\rho$ , the next order cannot be later than at the warehouse virtual time  $\gamma$  such that in the virtual time of retailer  $\rho$  the interval  $[\beta_i^\rho, \gamma]$  has length 1, that is  $\int_{\beta_i^\rho}^\gamma Density_\rho(y) dy = 1$ . So we choose as  $\beta_i^\rho$  the last warehouse ordering time before  $\gamma$  (Line 9–11).

---

**Pseudocode 1** Algorithm JMM

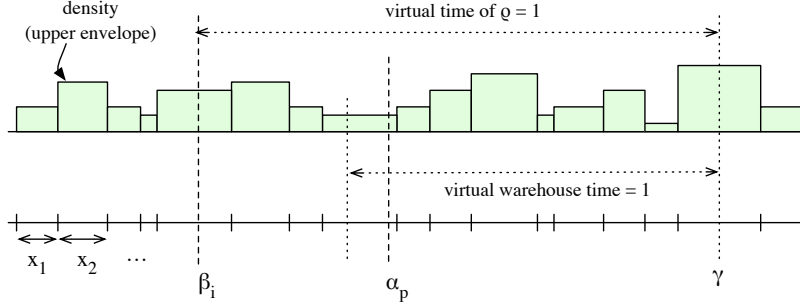
---

```

1:  $\hat{\mathbf{x}} \leftarrow \bar{0}$ ,  $\alpha_0 \leftarrow 0$ ,  $j \leftarrow 0$ ,  $v \leftarrow \sum_{t \in T} x_t$ 
2: while  $\alpha_j \leq v - 1$  do
3:    $\alpha_{j+1} \leftarrow \alpha_j + Rand()$ 
4:    $\hat{x}_{rt(\alpha_{j+1})} \leftarrow 1$ 
5:    $j \leftarrow j + 1$ 
6: for  $\rho = 1, \dots, m$  do
7:    $\beta_0^\rho \leftarrow 0$ ,  $i \leftarrow 0$ 
8:   while  $\int_{\beta_i^\rho}^v Density_\rho(y) dy \geq 1$  do
9:      $\gamma \leftarrow \min\{z : \int_{\beta_i^\rho}^z Density_\rho(y) dy = 1\}$ 
10:     $\beta_{i+1}^\rho \leftarrow \max\{\alpha_j : \alpha_j \leq \gamma\}$ 
11:     $\hat{x}_{rt(\beta_{i+1}^\rho)}^\rho \leftarrow 1$ 
12:     $i \leftarrow i + 1$ 
13: return  $\hat{\mathbf{x}}$ 

```

---



**Fig. 2.** An illustration of virtual times, the algorithm, and the proof of Lemma 4. Variables  $x_t$  are the lengths of the intervals on the warehouse virtual time axis. Variables  $x_t^\rho$  are represented by the areas of shaded rectangles. Function  $Density_\rho()$  is then represented by the upper envelope of these rectangles.

Let  $q$  be the number of warehouse orders generated. So their virtual times are  $\alpha_1, \alpha_2, \dots, \alpha_q$ , where  $v - 1 < \alpha_q < v$ . It could happen that we will set the same  $\hat{x}_t$  to 1 twice (that is, when  $rt(\alpha_j) = rt(\alpha_{j+1}) = t$ ), if  $x_t > 1/e$ , but not more than twice. This does not affect our analysis. The same comment applies to each retailer orders, of course.

**Lemma 1.** *Algorithm JMM terminates in polynomial time and it returns a feasible solution  $\hat{\mathbf{x}}$ .*

*Proof.* Let  $n = |D|$  be the total number of demands. The loop in Lines 2–5 terminates, because in each iteration it increases the current value of  $\alpha_j$  by at least  $1/e$ . Since  $v < |T|$ , this loop makes  $q \leq |T|/e \leq n/e$  iterations. Consider now the loop in Lines 8–12, for a fixed retailer  $\rho$ . Since  $1 = \int_{\beta_i^\rho}^\gamma Density_\rho(y) dy \leq \int_{\beta_i^\rho}^\gamma 1 dy = \gamma - \beta_i^\rho$ , we have  $\gamma \geq \beta_i^\rho + 1$ , so there exists a  $j$  for which  $\beta_i^\rho < \alpha_j \leq \gamma$ . Hence, by the choice of  $\beta_{i+1}^\rho$  in Line 10, we have  $\beta_{i+1}^\rho > \beta_i^\rho$  and the algorithm moves forward in each loop iteration. As  $(\beta_i^\rho)_i$  is a sub-sequence of  $(\alpha_j)_j$ , the number of these iterations is at most  $n/e$ . The integral in Line 9 is easy to compute incrementally in amortized constant time. So overall, the running time is  $O(nm)$ .

Finally, we have  $\int_{\beta_i^\rho}^{\beta_{i+1}^\rho} Density_\rho(y) dy \leq \int_{\beta_i^\rho}^\gamma Density_\rho(y) dy = 1$ . This implies that retailer  $\rho$  does not have a demand whose period is strictly contained in-between  $rt(\beta_i^\rho)$  and  $rt(\beta_{i+1}^\rho)$ , because this demand could not be fully served by the fractional solution  $\mathbf{x}$ . Therefore, placing orders for retailer  $\rho$  in both  $rt(\beta_i^\rho)$  and  $rt(\beta_{i+1}^\rho)$  suffices to satisfy any demand of  $\rho$  with period intersecting the interval  $[rt(\beta_i^\rho), rt(\beta_{i+1}^\rho)]$ .  $\square$

**Number of warehouse orders.** For proving the approximation guarantee of JMM, we first relate the number of warehouse orders in the solution returned

by the algorithm to  $v$ , the number of fractional warehouse orders in the optimal fractional solution  $\mathbf{x}$ .

By the choice of the random generator  $Rand()$ , the expected distance between two consecutive warehouse orders (measured in the virtual warehouse time) is  $\mathbf{E}[\alpha_{j+1} - \alpha_j] = (e-1)/e$ , and orders are placed while the condition in Line 2 holds. Intuitively, the number of warehouse  $\alpha$ 's is then around  $v \cdot e/(e-1)$ . To formalize this intuition, we use the following generalized version of Wald's identity (see, for example, [11]), customized to our need. Recall that, given a sequence  $(R_i)_{i \geq 1}$  of random variables, an integer-valued random variable  $N \geq 1$  is called a *stopping time* for  $(R_i)_{i \geq 1}$  if, for any  $k$ , the event " $N = k$ " is independent of  $R_j$  for  $j > k$ .

**Lemma 2.** *Let  $R_1, R_2, \dots$  be a sequence of (possibly dependent) random variables, where  $R_i \in [0, 1]$  and  $\mathbf{E}[R_i] \geq \xi$  for all  $i$ . Let  $N$  be a stopping time for this sequence with finite expectation. Then,  $\mathbf{E}[\sum_{i=1}^N R_i] \geq \xi \cdot \mathbf{E}[N]$ . Both inequalities can be replaced by equalities.*

**Lemma 3.** *The expected number of warehouse orders generated by JMM is at most  $\frac{e}{e-1} \cdot v$ , where  $v$  is the number of warehouse orders in the optimal fractional solution  $\mathbf{x}$ .*

*Proof.* The random variables  $R_j = \alpha_j - \alpha_{j-1}$ , for  $j = 1, \dots, q$ , satisfy conditions of Lemma 2 with  $\xi = 1 - 1/e$  and with the stopping time  $N = q$  defined as the minimum index for which  $\alpha_q \geq v - 1$ . Then,

$$\begin{aligned} (1 - 1/e) \cdot \mathbf{E}[q] &= \mathbf{E}[\sum_{j=1}^q R_j] \\ &= \mathbf{E}[R_q + \sum_{j=1}^{q-1} R_j] \\ &\leq 1 + (v - 1) = v, \end{aligned}$$

We thus obtain  $\mathbf{E}[q] \leq \frac{e}{e-1} \cdot v$ . □

**Number of retailer orders.** To obtain an analogous bound on the number of retailer  $\rho$ 's orders, we need to estimate the frequency of  $\rho$ 's orders in  $\hat{\mathbf{x}}$ . Analogously, to apply Wald's identity, we need to show that when considering retailer  $\rho$  in the algorithm (cf. Line 6 and the following), the expected differences between consecutive  $\beta^\rho$ 's are large enough, i.e., that  $\mathbf{E}[\int_{\beta_i^\rho}^{\beta_{i+1}^\rho} \text{Density}_\rho(y) dy] \geq (e-1)/e$ . However, for the reason we pin down in the proof of Lemma 4, this relation might not be true for the last value of  $\beta^\rho$ . To alleviate this problem, we make the following thought experiment: we consider a modified version of the algorithm in which the condition  $\alpha_j \leq v - 1$  in Line 2 is replaced by  $\alpha_j \leq v$ . This may incur at most two additional loop iterations in Lines 2–5 (calls of  $Rand()$  producing at most two more warehouse orders). As  $\beta^\rho$  is a deterministic function of the sequence  $\alpha$ , the newly produced sequence of  $\beta^\rho$  (denoted  $\tilde{\beta}^\rho$ ) is identical to the original one, except that the last retailer order might be issued a little later. Thus for analyzing the (expected) number of such orders, we may analyze sequence  $\tilde{\beta}^\rho$  instead of  $\beta^\rho$ .

**Lemma 4.** For any retailer  $\rho$  and two consecutive values  $\bar{\beta}_i^\rho$  and  $\bar{\beta}_{i+1}^\rho$ , it holds that

$$\mathbf{E} \left[ \int_{\bar{\beta}_i^\rho}^{\bar{\beta}_{i+1}^\rho} \text{Density}_\rho(y) dy \right] \geq (e - 1)/e.$$

*Proof.* Consider the value of  $\gamma$  computed in Line 9 in the same iteration which computes the value of  $\bar{\beta}_{i+1}^\rho$ . Since

$$\int_{\bar{\beta}_i^\rho}^{\bar{\beta}_{i+1}^\rho} \text{Density}_\rho(y) dy = 1 - \int_{\bar{\beta}_{i+1}^\rho}^{\gamma} \text{Density}_\rho(y) dy \geq 1 - [\gamma - \bar{\beta}_{i+1}^\rho], \quad (3)$$

it is sufficient to show that  $\mathbf{E}[\gamma - \bar{\beta}_{i+1}^\rho] \leq 1/e$ . Define  $p$  to be the smallest index for which  $\alpha_p > \gamma - 1$  (cf. Figure 2).

If  $\gamma \leq \alpha_p + 1/e$  then  $\bar{\beta}_{i+1}^\rho = \alpha_p$  with probability 1 (by the definition of  $\text{Rand}()$ ), so  $\mathbf{E}[\gamma - \bar{\beta}_{i+1}^\rho] \leq 1/e$  is trivial. Thus, in the rest of the proof we can assume that  $\gamma \geq \alpha_p + 1/e$ .

We now focus on the distance  $\mu = \gamma - \alpha_p$  and consider the random choice of the next warehouse order moment  $\alpha_{p+1}$ , which was computed by the algorithm as  $\alpha_{p+1} \leftarrow \alpha_p + y$ , where  $y$  is the random variable of this particular execution of  $\text{Rand}()$ . (Note that our modification of the sequence  $\alpha$  guarantees that  $\alpha_p$  is not the last order, and hence  $\alpha_{p+1}$  is well-defined.)

By the case assumption we have  $1/e < \mu < 1$ . In case  $\alpha_{p+1} > \gamma$ , which is equivalent to  $y > \mu$ , the next retailer  $\rho$ 's order is placed in  $\bar{\beta}_{i+1}^\rho = \alpha_p$ . Then  $\gamma - \bar{\beta}_{i+1}^\rho = \mu$ . In the remaining case, we have  $\alpha_{p+1} \leq \gamma$ , which is equivalent to  $y \leq \mu$ . Then  $\bar{\beta}_{i+1}^\rho \geq \alpha_{p+1}$ , so  $\gamma - \bar{\beta}_{i+1}^\rho \leq \gamma - \alpha_{p+1} = \gamma - (\alpha_p + y) = \mu - y$ . Hence:

$$\begin{aligned} \mathbf{E}[\gamma - \bar{\beta}_{i+1}^\rho] &\leq \int_{1/e}^{\mu} (\mu - y)y^{-1} dy + \int_{\mu}^1 \mu y^{-1} dy \\ &= \mu \int_{1/e}^1 y^{-1} dy - \int_{1/e}^{\mu} 1 dy = 1/e, \end{aligned}$$

thus completing the proof.  $\square$

**Lemma 5.** For any retailer  $\rho$ , the expected number of its orders generated by JMM is at most  $\frac{e}{e-1} \cdot v^\rho$ , where  $v^\rho$  is the number of retailer orders in the optimal fractional solution  $\mathbf{x}$ .

*Proof.* For the sequence  $\beta_1^\rho, \beta_2^\rho, \dots, \beta_s^\rho$  generated by the algorithm, it holds that

$$\int_0^{\beta_{s-1}^\rho} \text{Density}_\rho(y) dy = \int_0^v \text{Density}_\rho(y) dy - \int_{\beta_{s-1}^\rho}^v \text{Density}_\rho(y) dy \leq v^\rho - 1,$$

where the inequality follows by the while condition in Line 8. Consider now the modified sequence  $\bar{\beta}^\rho = \bar{\beta}_1^\rho, \bar{\beta}_2^\rho, \dots, \bar{\beta}_s^\rho$ . Recall that  $\beta_i^\rho = \bar{\beta}_i^\rho$  for  $i < s$  and  $\bar{\beta}_s^\rho \geq \beta_s^\rho$ . We may now apply Wald's identity: the random variables  $R_j = \int_{\beta_{j-1}^\rho}^{\beta_j^\rho} \text{Density}_\rho(y) dy$  for  $j = 1, \dots, s$  satisfy conditions of Lemma 2 with  $\xi \geq 1 - 1/e$

and with the stopping time  $N = s$  defined as the minimum index for which  $\int_0^{\beta_s} \text{Density}_\rho(y) dy > v^\rho - 1$ . Then,

$$\begin{aligned} (1 - 1/e) \cdot \mathbf{E}[s] &\leq \mathbf{E}[\sum_{j=1}^s R_j] \\ &= \mathbf{E}[R_s + \sum_{j=1}^{s-1} R_j] \\ &\leq 1 + (v^\rho - 1) = v^\rho, \end{aligned}$$

We thus obtain  $\mathbf{E}[s] \leq \frac{e}{e-1} \cdot v^\rho$ .

**Approximation guarantee.** By Lemma 1, Algorithm JMM returns a feasible solution  $\hat{\mathbf{x}}$ . Furthermore, by Lemma 3 and Lemma 5 the expected number of orders (respectively, warehouse and retailer ones) it produces is at most  $\frac{e}{e-1}$  times the number of orders in the optimal fractional solution  $\mathbf{x}$ . Hence, we obtain the following bound.

**Theorem 1.** *Algorithm JMM is a polynomial-time randomized  $\frac{e}{e-1}$ -approximation algorithm for JRP-D.*

### 3 APX-Hardness

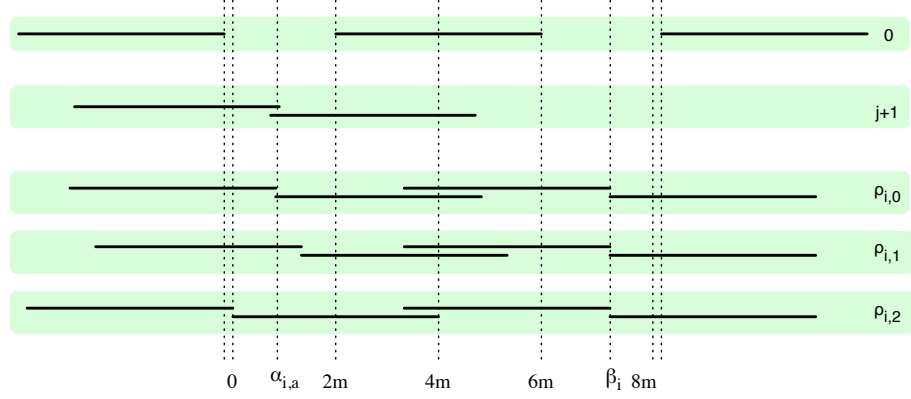
Let JRP-D<sub>E4</sub> be the restriction of JRP-D where each retailer has at most four demands and all demand intervals are of the same length. We show that JRP-D<sub>E4</sub> is APX-hard, using the result by Alimonti and Kann [1] that Vertex Cover is APX-hard even for cubic graphs. Roughly speaking, on the basis of any cubic graph  $G = (V, E)$  with  $n$  vertices (where  $n$  is even) and  $m = 1.5n$  edges, in polynomial time we construct an instance  $\mathcal{J}_G$  of JRP-D<sub>E4</sub>, such that the existence of a vertex cover for  $G$  of size at most  $K$  is equivalent to the existence of an order schedule for  $\mathcal{J}_G$  of cost at most  $10.5n + K + 6$  (and such equivalence can be shown by polynomial-time mappings between solutions).

**Construction of instance  $\mathcal{J}_G$ .** In the following, we fix a cubic graph  $G$  with vertices  $v_0, \dots, v_{n-1}$  and edges  $e_0, \dots, e_{m-1}$ . We construct  $\mathcal{J}_G$  consisting of  $1 + m + 3n$  retailers, where all the ordering costs are 1, that is  $L = 1$  and  $b_\rho = 1$  for all  $\rho$ 's. Each retailer has at most four demands and all demand periods are of length  $4m$ .

With each edge  $e_j$  we will associate a pair of time points  $2j, 2j + 1$ , each corresponding to an endpoint of  $e_j$ . For each vertex  $v_i$ , let  $e_{j_0(i)}, e_{j_1(i)}$  and  $e_{j_2(i)}$  be the three edges incident to  $v_i$ . Vertex  $v_i$  will be associated with four time points. One of these points is  $\beta_i = 8m - i$ , and the other three points are those that represent  $v_i$  in the point pairs corresponding to edges. Specifically, for  $a \in \{0, 1, 2\}$ , if  $e_{j_a(i)} = (v_i, v_{i'})$  then let  $\alpha_{i,a} = 2j_a(i)$  if  $i < i'$ , otherwise let  $\alpha_{i,a} = 2j_a(i) + 1$ . These three other points are then  $\alpha_{i,0}, \alpha_{i,1}$  and  $\alpha_{i,2}$ .

$\mathcal{J}_G$  will consist of  $1 + n + m$  gadgets: a *vertex gadget* VG <sub>$i$</sub>  for each vertex  $v_i$ , an *edge gadget* EG <sub>$j$</sub>  for each edge  $e_j$ , and one special *support gadget* SG. We now describe these gadgets (see also Figure 3 for reference).





**Fig. 3.** The construction of instance  $\mathcal{J}_G$ . The figure shows the support gadget SG, an edge gadget  $\text{EG}_j$  and a vertex gadget  $\text{VG}_i$ . Shaded regions represent retailers. Thick horizontal intervals represent demand periods.

**The support gadget.** SG consists simply of one retailer 0 with three demands  $(0, -4m - 1, -1)$ ,  $(0, 2m, 6m)$  and  $(0, 8m + 1, 12m + 1)$ . Retailer 0 does not have any other demands. Since the demand periods of retailer 0 are disjoint, they will require three orders. We will show later that these orders can be assumed to be at times  $-1$ ,  $4m$  and  $8m + 1$ .

**Edge gadgets.** For any edge  $e_j$ ,  $\text{EG}_j$  consists of retailer  $j + 1$  with demands  $(j + 1, 2j + 1 - 4m, 2j + 1)$  and  $(j + 1, 2j, 2j + 4m)$ . Retailers  $1, 2, \dots, m$  will not contain any other demands. The demands of each such retailer  $j + 1$  can be satisfied with one order in the interval  $[2j, 2j + 1]$ . If there is no such order, two orders are needed.

**Vertex gadgets.** For any vertex  $v_i$ ,  $\text{VG}_i$  will involve three retailers  $\rho_{i,a} = m + 1 + 3i + a$ , for  $a = 0, 1, 2$ , and their demands. Each retailer  $\rho_{i,a}$  will have four demands:

$$(\rho_{i,a}, \alpha_{i,a} - 4m, \alpha_{i,a}), (\rho_{i,a}, \alpha_{i,a}, \alpha_{i,a} + 4m), \\ (\rho_{i,a}, \beta_i - 4m, \beta_i), \text{ and } (\rho_{i,a}, \beta_i, \beta_i + 4m).$$

For ease of reference, we will denote the periods of these four demands as  $Q_{i,a}^0$ ,  $Q_{i,a}^1$ ,  $Q_{i,a}^2$  and  $Q_{i,a}^3$ , in the order listed above. Note that  $Q_{i,a}^0 \cap Q_{i,a}^1 = \{\alpha_{i,a}\}$ ,  $Q_{i,a}^1 \cap Q_{i,a}^2 = [\beta_i - 4m, \alpha_{i,a} + 4m] \neq \emptyset$ ,  $Q_{i,a}^2 \cap Q_{i,a}^3 = \{\beta_i\}$ , and that other pairs of demand periods of retailer  $\rho_{i,a}$  are disjoint. The important property of retailer  $\rho_{i,a}$  is that it requires two orders, and that the only way to satisfy his demands by two orders is when these orders are at  $\alpha_{i,a}$  and  $\beta_i$ . There are infinitely many ways to satisfy  $\rho_{i,a}$ 's demands with three orders. In particular, orders at times

$-1$ ,  $4m$  and  $8m + 1$  work, and these times will be used in our construction for this purpose.

**Lemma 6.** *If  $G$  has a vertex cover  $U$  of size  $K$ , then there is an order schedule for  $\mathcal{J}_G$  of cost at most  $10.5n + K + 6$ .*

*Proof.* To construct a schedule for  $\mathcal{J}_G$ , we first include warehouse orders at times  $-1$ ,  $4m$  and  $8m + 1$ , and each of them is joined by retailer 0 (of gadget SG). This incurs the cost of 6.

Next, consider a vertex  $v_i$  and the associated gadget  $VG_i$ . If  $v_i \notin U$ , have retailers  $\rho_{i,0}$ ,  $\rho_{i,1}$  and  $\rho_{i,2}$  join warehouse orders at times  $-1$ ,  $4m$  and  $8m + 1$ . The additional cost for  $VG_i$  is then 9. If  $v_i \in U$ , for each  $i = 0, 1, 2$  we make two orders from retailer  $\rho_{i,a}$ , one at  $\alpha_{i,a}$  and one at  $\beta_i$ . Since the warehouse order at  $\beta_i$  is shared between these three retailers, the additional cost for  $VG_i$  is  $1 + 3(1 + 2) = 10$ . As  $U$  is a vertex cover, the choices we made for vertex gadgets imply that, for each edge  $e_j$ , there is now a warehouse order at a time  $t \in \{2j, 2j + 1\}$ , that corresponds to a vertex that covers  $e_j$ . We can thus have retailer  $j + 1$  of  $EG_j$  join this order at cost 1. Thus, the total cost is  $6 + 9(n - K) + 10K + m = 10.5n + K + 6$ , as required.  $\square$

**Recovering a vertex cover from an order schedule.** We now want to show that from any order schedule of cost  $10.5n + K + 6$  for  $\mathcal{J}_G$  we can compute a vertex cover of  $G$  of size  $K$ . This part takes some additional work, as we would like to assume that the given order schedule  $S$  is in the following normal form:

- (nf1)  $S$  has warehouse orders at times  $-1$ ,  $4m$  and  $8m + 1$ , and these orders are used by retailer 0.
- (nf2) For each edge gadget  $EG_j$ ,  $S$  has an order at a time  $t \in \{2j, 2j + 1\}$  that involves retailer  $j + 1$ .
- (nf3) For each vertex gadget  $VG_i$ , one of the following two conditions holds:
  - $S$  has orders of retailer  $\rho_{i,a}$  at times  $\beta_i$  and  $\alpha_{i,a}$ , for each  $a = 0, 1, 2$ .
  - $S$  has orders from retailer  $\rho_{i,a}$  at times  $-1$ ,  $4m$  and  $8m + 1$ , and there are no warehouse orders at times  $\beta_i$  and  $\alpha_{i,a}$ , for each  $a = 0, 1, 2$ .
- (nf4) There are no other orders in  $S$ .

The following lemma states that we may assume an order schedule to be in the normal form, without loss of generality.

**Lemma 7.** *For any order schedule  $S$  for  $\mathcal{J}_G$ , we can compute a normal-form schedule  $S'$  whose cost is at most the cost of  $S$ .*

*Proof.* We prove this claim by gradually modifying  $S$  to obtain  $S'$ . We can assume that orders occur only at endpoints of demand intervals. We can also assume that the first order is at the earliest deadline and the last order is at the latest release time. Therefore we can assume that  $S$  has orders at times  $-1$  and  $8m + 1$ . Because of demand  $(0, 2m, 6m)$ , there must be an order between  $2m$  and  $6m$ , let's say at time  $t$ . There are no demand intervals ending in  $[2m, 4m)$ , so if  $2m \leq t < 4m$ , we can shift the order at  $t$  rightward to  $4m$ . Similarly, there

are no demand intervals starting in  $(4m, 6m]$ , so if  $4m < t \leq 6m$  then we can shift the order at  $t$  leftward to  $4m$ . This does not increase the cost. This way we will have an order at time  $4m$  and no other orders in the interval  $[2m, 6m]$ . The resulting schedule satisfies [Property \(nf1\)](#).

Next, consider any edge gadget  $EG_j$ . If retailer  $j + 1$  is not involved in an order at time  $2j$  or  $2j + 1$ , then it must be involved in two orders, one before  $2j$  and the other after  $2j + 1$ . We can remove him from these two orders and place an order at time  $2j$  (or  $2j + 1$ ) without increasing the cost. The resulting schedule satisfies [Property \(nf2\)](#).

Finally, consider any vertex gadget  $VG_i$ . Suppose that  $S$  has a warehouse order at time  $\beta_i$ . Let  $a \in \{0, 1, 2\}$ . We can assume that retailer  $\rho_{i,a}$  participates in the order at time  $\beta_i$ , since otherwise we can replace his order at any later time (there must be one in the demand period  $Q_{i,a}^3$ ) by the order at  $\beta_i$ . Suppose that  $S$  does not have an order from retailer  $\rho_{i,a}$  at time  $\alpha_{i,a}$ . Then retailer  $\rho_{i,a}$  must participate in at least two orders in addition to that at  $\beta_i$ . We can remove him from these two orders and add an order from  $\rho_{i,a}$  at time  $\alpha_{i,a}$  (or have  $\rho_{i,a}$  join the warehouse order at time  $\alpha_{i,a}$  if it already exists). This operation does not increase the cost. We have thus shown that if there is a warehouse order at time  $\beta_i$  then, without loss of generality, we can assume that each retailer  $\rho_{i,a}$ , for  $a = 0, 1, 2$ , has orders at times  $\alpha_{i,a}$  and  $\beta_i$ , and no other orders.

Suppose that  $S$  does not have an order at time  $\beta_i$ , and that for some  $a \in \{0, 1, 2\}$  it has an order at time  $\alpha_{i,a}$  that does not involve  $\rho_{i,a}$ . In this case  $\rho_{i,a}$  must be in three orders. We can remove him from these orders, add him to the warehouse order at  $\alpha_{i,a}$ , and add a new order  $\beta_i$ . The cost of the added orders is at most 3, so the overall cost will not increase. By the previous paragraph, we can then assume that for each of the other two retailers  $\rho_{i,a'}$  of  $VG_i$ ,  $a' \neq a$ , he also has exactly orders, at  $\alpha_{i,a'}$  and  $\beta_i$ .

The two paragraphs above show that either we can modify  $S$  so that the orders from  $VG_i$  satisfy the first condition in [Property \(nf3\)](#), or there are no warehouse orders at times  $\beta_i$ ,  $\alpha_{i,0}$ ,  $\alpha_{i,1}$  and  $\alpha_{i,2}$ . In the latter case though, each retailer  $\rho_{i,0}$ ,  $\rho_{i,1}$ ,  $\rho_{i,2}$  must be in three orders, which, without loss of generality, are those at times  $-1$ ,  $4m$  and  $8m + 1$ , i.e.,  $VG_i$  satisfies the second condition in [Property \(nf3\)](#).

As the orders described in the proof above are the only necessary orders, [Property \(nf4\)](#) holds as well and the lemma follows.  $\square$

**Lemma 8.** *On the basis of an order schedule  $S$  for  $\mathcal{J}_G$  of cost  $10.5n + K + 6$ , we can construct in polynomial time a vertex cover of  $G$  of size  $K$ .*

*Proof.* By [Lemma 7](#), we can assume that  $S$  is in the normal form. For such schedules it is easy to compute the cost. Suppose that  $S$  has  $\ell$  orders in the interval  $(6m, 8m]$ , that is at times  $\beta_i$ . For each  $v_i$  for which  $S$  has an order at  $\beta_i$ , we pay 1 for the warehouse cost at  $\beta_i$ , for each edge  $e_{j_a(i)}$  incident to  $v_i$  retailer  $\rho_{i,a}$  pays cost of 2 and we pay 1 more for the warehouse cost at  $\alpha_{i,a}$ . So the total cost associated with such  $v_i$ 's is  $10\ell$ . For each  $i$  for which  $S$  does not have an order at  $\beta_i$  we pay 9 for the retailer cost (3 for each  $\rho_{i,a}$ ). So the cost

associated with such  $v_i$ 's is  $9(n - \ell)$ . We then have additional cost  $m = 1.5n$  for the retailer cost of the gadgets  $EG_j$ , plus 3 for the warehouse cost at  $-1$ ,  $4m$ ,  $8m + 1$  and 3 for the retailer cost for gadget  $SG$  at these times. This gives us total cost  $10.5n + \ell + 6$ , and we can conclude that we must have  $\ell = K$ .

Define now  $U$  to be the set of those vertices  $v_i$  for which  $S$  makes an order at time  $\beta_i$  from the retailers in  $VG_i$ . Consider some edge  $e_j$ . Using the normal form, we first obtain that there is an order at time  $2j$  or  $2j+1$ , say at  $2j$ . Choose  $i$  and  $a$  for which  $2j = \alpha_{i,a}$ . By the normal form, gadget  $VG_i$  satisfies the first condition in [Property \(nf3\)](#), which in turn implies that  $v_i \in U$ . This shows that each edge  $e_j$  is covered by  $U$ , so  $G$  has a vertex cover of size  $K$ .  $\square$

**Theorem 2.** *Problem  $JRP-D_{E4}$ , that is the restriction of  $JRP-D$  to instances with equal demand intervals and at most four demands per retailer, is  $\text{APX-hard}$ .*

*Proof.* By [1], there exists an  $\epsilon > 0$ , such that there is no polynomial-time  $(1 + \epsilon)$ -approximation algorithm for the vertex cover problem in cubic graphs, unless  $\mathbb{P} = \text{NP}$ . Towards a contradiction, assume that we can compute in polynomial-time a  $(1 + \epsilon/24)$ -approximation for  $JRP-D_{E4}$ . Fix any cubic graph  $G$  of  $n \geq 6$  vertices and  $m$  edges with a vertex cover of size  $K$  and construct an instance  $\mathcal{J}_G$ . By [Lemma 6](#),  $\mathcal{J}_G$  has an order schedule of cost at most  $10.5n + K + 6$ , so our approximation algorithm finds a schedule of cost  $C \leq (1 + \epsilon/24) \cdot (10.5n + K + 6)$ . As the graph is cubic,  $K \geq m/3 = n/2$ , and therefore  $10.5n + 6 \leq 11.5n \leq 23K$ . Thus,  $C \leq 10.5n + (1 + \epsilon)K + 6$ . (In fact, as all the costs are integers, the obtained order schedule has cost at most  $10.5n + \lfloor (1 + \epsilon)K \rfloor + 6$ .) Finally, we may use the construction of [Lemma 8](#) to compute a vertex cover for  $G$  of size at most  $\lfloor (1 + \epsilon)K \rfloor$ , which contradicts the result of [1].  $\square$

## 4 1.5-Approximation for Equal-Length Demand Periods

In this section, we present a 1.5-competitive algorithm for the case where all the demand periods are of equal length. Without loss of generality, we will assume that the length of each period is 1.

We denote the input instance by  $\mathcal{I}$ . Let the *width* of an instance be the difference between the deadline of the last demand and the release time of the first one. The building block of our approach is an algorithm that creates an optimal solution to an instance of width at most 3. Later, we divide  $\mathcal{I}$  into overlapping sub-instances of width 3, solve each of them optimally, and finally show that by aggregating their solutions we obtain a 1.5-approximation for  $\mathcal{I}$ .

**Lemma 9.** *A solution to an instance  $\mathcal{J}$  of width at most 3 consisting of unit-length demand periods can be computed in polynomial time.*

*Proof.* We shift all demands in time, so that  $\mathcal{J}$  is entirely contained in interval  $[0, 3]$ . Recall that  $L$  is the warehouse ordering cost and  $b_\rho$  is the ordering cost of retailer  $\rho \in \{1, 2, \dots, m\}$ . Without loss of generality, we can assume that all retailers  $1, \dots, m$  have some demands.

Let  $d_{\min}$  be the first deadline of a demand from  $\mathcal{J}$  and  $r_{\max}$  the last release time. If  $r_{\max} \leq d_{\min}$ , then placing one order at any time from  $[r_{\max}, d_{\min}]$  is sufficient (and necessary). Its cost is then equal to  $L + \sum_{\rho} b_{\rho}$ .

Thus, in the following we focus on the case  $d_{\min} < r_{\max}$ . Any feasible solution has to place an order at or before  $d_{\min}$  and at or after  $r_{\max}$ . Furthermore, by shifting these orders we may assume that the first and last orders occur exactly at times  $d_{\min}$  and  $r_{\max}$ , respectively.

The problem is thus to choose a set  $T$  of warehouse ordering times that contains  $d_{\min}$ ,  $r_{\max}$ , and possibly other times from the interval  $(d_{\min}, r_{\max})$ , and then to decide, for each retailer  $\rho$ , which warehouse orders it joins. Note that  $r_{\max} - d_{\min} \leq 1$ , and therefore each demand period contains  $d_{\min}$ ,  $r_{\max}$ , or both. Hence, all demands of a retailer  $\rho$  can be satisfied by joining the warehouse orders at times  $d_{\min}$  and  $r_{\max}$  at additional cost of  $2b_{\rho}$ . It is possible to reduce the retailer ordering cost to  $b_{\rho}$  if (and only if) there is a warehouse order that occurs within  $D_{\rho}$ , where  $D_{\rho}$  is the intersection of all demand periods of retailer  $\rho$ . (To this end,  $D_{\rho}$  has to be non-empty.)

Hence, the optimal cost for  $\mathcal{J}$  can be expressed as the sum of four parts:

- (i) the unavoidable ordering cost  $b_{\rho}$  for each retailer  $\rho$ ,
- (ii) the additional ordering cost  $b_{\rho}$  for each retailer  $\rho$  with empty  $D_{\rho}$ ,
- (iii) the total warehouse ordering cost  $L \cdot |T|$ , and
- (iv) the additional ordering cost  $b_{\rho}$  for each retailer  $\rho$  whose  $D_{\rho}$  is non-empty and does not contain any ordering time from  $T$ .

As the first two parts of the cost are independent of  $T$ , we focus on minimizing the sum of parts (iii) and (iv) that we call the *adjusted cost*. Let  $C(x)$  be the minimum possible adjusted cost “in the interval  $[d_{\min}, x]$ ” under the assumption that there is an order at time  $x$ . Formally,  $C(x)$  is the minimum, over all choices of sets  $T \subseteq [d_{\min}, x]$  that contain  $d_{\min}$  and  $x$ , of  $L \cdot |T| + \sum_{\rho \in Q(T)} b_{\rho}$ , where  $Q(T)$  is the set of retailers  $\rho$  for which  $D_{\rho} \neq \emptyset$  and  $D_{\rho} \subseteq [0, x] - T$ . (Note that the second term consists of expenditures that actually occur outside the interval  $[d_{\min}, x]$ .)

As there are no  $D_{\rho}$ ’s strictly to the left of  $d_{\min}$ ,  $C(d_{\min}) = L$ . Furthermore, to compute  $C(x)$  for any  $x \in (d_{\min}, r_{\max}]$ , we can express it recursively using the value of  $C(y)$  for  $y \in [d_{\min}, x)$  being the warehouse order time immediately preceding  $x$  in the set  $T$  that realizes  $C(x)$ . This gives us the formula

$$C(x) = L + \min_{y \in [d_{\min}, x)} \left( C(y) + \sum_{\rho: \emptyset \neq D_{\rho} \subset (y, x)} b_{\rho} \right).$$

In the minimum above, we may restrict computation of  $C(x)$  to  $x$ ’s and  $y$ ’s that are ends of demand periods. Hence, the actual values of function  $C(\cdot)$  can be computed by dynamic programming in polynomial time. Finally, the total adjusted cost is equal to  $C(r_{\max})$ . Once we computed the minimum adjusted cost, recovering the actual orders can be performed by a straightforward extension of the dynamic programming presented above.  $\square$

Now, we construct an approximate solution for the original instance  $\mathcal{I}$  consisting of unit-length demand periods. For  $i \in \mathbb{N}$ , let  $\mathcal{I}_i$  be the sub-instance containing all demands entirely contained in  $[i, i + 3)$ . By [Lemma 9](#), an optimal solution for  $\mathcal{I}_i$ , denoted  $A(\mathcal{I}_i)$ , can be computed in polynomial time. Let  $S_0$  be the solution created by aggregating  $A(\mathcal{I}_0), A(\mathcal{I}_2), A(\mathcal{I}_4), \dots$  and  $S_1$  by aggregating  $A(\mathcal{I}_1), A(\mathcal{I}_3), A(\mathcal{I}_5), \dots$ . Among solutions  $S_0$  and  $S_1$ , we output the one with the smaller cost.

**Theorem 3.** *The above algorithm produces a feasible solution of cost at most 1.5 times the optimum cost.*

*Proof.* Each unit-length demand of instance  $\mathcal{I}$  is entirely contained in some  $\mathcal{I}_{2k}$  for some  $k \in \mathbb{N}$ . Hence, it is satisfied in  $A(\mathcal{I}_{2k})$ , and thus also in  $S_0$ , which yields the feasibility of  $S_0$ . An analogous argument shows the feasibility of  $S_1$ .

To estimate the approximation ratio, we fix an optimal solution  $\text{OPT}$  for instance  $\mathcal{I}$  and let  $\text{opt}_i$  be the cost of  $\text{OPT}$ 's orders in the interval  $[i, i + 1)$ . Note that  $\text{OPT}$ 's orders in  $[i, i + 3)$  satisfy all demands contained entirely in  $[i, i + 3)$ . Since  $A(\mathcal{I}_i)$  is an optimal solution for these demands, we have  $\text{cost}(A(\mathcal{I}_i)) \leq \text{opt}_i + \text{opt}_{i+1} + \text{opt}_{i+2}$  and, by taking the sum, we obtain  $\text{cost}(S_0) + \text{cost}(S_1) \leq \sum_i \text{cost}(A(\mathcal{I}_i)) \leq 3 \cdot \text{cost}(\text{OPT})$ . Therefore, either of the two solutions ( $S_0$  or  $S_1$ ) has cost at most  $1.5 \cdot \text{cost}(\text{OPT})$ .  $\square$

## 5 Final Comments

The main result we presented is an  $e/(e-1)$ -approximation algorithm for JRP-D. The main open question is of course whether and by how much the approximation ratio for JRP-D can be improved, and what ratio can be achieved with a deterministic algorithm. A careful reader might have noticed that some improvements should be still possible: since the density function of the next order used in our algorithm is non-zero starting at  $1/e < 0.5$  (in terms of the fractional solution “mass”), for time points that are farther than  $2/e$  and not farther than 1 from the last warehouse order, with some small probability our algorithm gets a second chance to use them for the next order. We have not pursued this approach, since the improvement in the ratio will be very minor and its analysis looks tedious.

There is a simple algorithm for JRP-D that provides a  $(1, 2)$ -approximation, in the following sense: its warehouse order cost is not larger than that in the optimum, while its retailer order cost is at most twice that in the optimum [\[10\]](#). One can then try to balance the two approaches by choosing each algorithm with a certain probability. This simple approach does not improve the ratio. But it may be possible to achieve a better ratio if, instead of using our algorithm as presented, we appropriately adjust the probability distribution.

If we parametrize JRP-D by the maximum number  $p$  of demand periods of each retailer, its complexity status is essentially resolved: for  $p \geq 3$  the problem is  $\text{APX-hard}$  [\[10\]](#), while for  $p \leq 2$  it can be solved in polynomial time (by a greedy algorithm for  $p = 1$  and dynamic programming for  $p = 2$ ). In case of equal-length

demand periods, we showed that the problem is  $\mathsf{APX}$ -hard for  $p \geq 4$ , but the case  $p = 3$  remains open, and it would be nice to settle this case as well. We conjecture that in this case the problem is  $\mathsf{NP}$ -complete.

*Acknowledgements.* We would like to thank Łukasz Jeż, Jiri Sgall, Grzegorz Stachowiak, and Neal Young for stimulating discussions and useful comments.

## References

1. P. Alimonti and V. Kann. Some  $\mathsf{APX}$ -completeness results for cubic graphs. *Theoretical Computer Science*, 237(1–2):123–134, 2000.
2. E. Arkin, D. Joneja, and R. Roundy. Computational complexity of uncapacitated multi-echelon production planning problems. *Operations Research Letters*, 8(2):61–66, 1989.
3. L. Becchetti, A. Marchetti-Spaccamela, A. Vitaletti, P. Korteweg, M. Skutella, and L. Stougie. Latency-constrained aggregation in sensor networks. *ACM Transactions on Algorithms*, 6(1):13:1–13:20, 2009.
4. N. Buchbinder, T. Kimbrel, R. Levi, K. Makarychev, and M. Sviridenko. Online make-to-order joint replenishment model: primal dual competitive algorithms. In *Proc. of the 19th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 952–961, 2008.
5. S. Khanna, J. Naor, and D. Raz. Control message aggregation in group communication protocols. In *Proc. of the 29th Int. Colloq. on Automata, Languages and Programming (ICALP)*, pages 135–146, 2002.
6. R. Levi, R. Roundy, and D. B. Shmoys. A constant approximation algorithm for the one-warehouse multi-retailer problem. In *Proc. of the 16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 365–374, 2005.
7. R. Levi, R. Roundy, and D. B. Shmoys. Primal-dual algorithms for deterministic inventory problems. *Mathematics of Operations Research*, 31(2):267–284, 2006.
8. R. Levi, R. Roundy, D. B. Shmoys, and M. Sviridenko. A constant approximation algorithm for the one-warehouse multiretailer problem. *Management Science*, 54(4):763–776, 2008.
9. R. Levi and M. Sviridenko. Improved approximation algorithm for the one-warehouse multi-retailer problem. In *Proc. of the 9th Int. Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, pages 188–199, 2006.
10. T. Nonner and A. Souza. Approximating the joint replenishment problem with deadlines. *Discrete Mathematics, Algorithms and Applications*, 1(2):153–174, 2009.
11. Wikipedia. Wald’s identity. [http://en.wikipedia.org/wiki/Wald's\\_equation](http://en.wikipedia.org/wiki/Wald's_equation), 2012.